Multimodal Modular Meta-Learning

Harshvardhan D. Sikka
Georgia Institute of Technology
Manifold Computing
OpenMined
harsh@manifoldcomputing.com

Atharva A. Tendle
University of Nebraska-Lincoln
Manifold Computing
OpenMined
atharva.tendle@huskers.unl.edu

Amr Kayid
German University in Cairo
Manifold Computing
OpenMined
amrmkayid@gmail.com

Abstract

Many real world prediction problems involve structured tasks across multiple modalities. We propose to extend previous work in modular meta learning to the multimodal setting. Specifically, we present an algorithmic approach to apply task aware modulation to a modular meta learning system that decomposes structured multimodal problems into a set of modules that can be reassembled to learn new tasks. We also propose a series of experiments to compare this approach with state of the art modular and multimodal meta learning approaches on multimodal function prediction and image classification tasks.

1 Introduction

Leveraging previous experiences to acquire new skills is relatively easy for humans, but it presents significant theoretical and computational challenges for machine learning systems. Current machine learning systems are "specialists" which excel in the tasks they are trained for but often fall apart when attempting a different task. Meta-learning is a field that has the potential to create "generalist" algorithms, and it does so through the process of "learning to learn" [1, 2]. These "generalist" algorithms can adapt to newer tasks by leveraging prior experiences. In recent years there have been advancements that allow machines to adapt to new tasks by learning an internal representation over the tasks in the training-data distribution [3–6]. These advancements have achieved some degree of success in unimodal domains, and some have been extended to the multimodal setting. For example, MMAML [7] attempts to solve the multi-modal problem by creating a framework that identifies the mode of sampled tasks and then modulates meta-learned priors that better fit the mode.

A wide variety of interesting and relevant real world problems demonstrate multimodality along with some inherent structure, including robotics tasks, autonomous navigation, and multimodal language and vision tasks. Many model agnostic meta learning approaches will intuitively struggle in these domains due to the constraints of the learning algorithms and biases they employ to generalize to new tasks. For example, in the MMAML methodology introduced in [7] the structure of the architecture stays constant and therefore the performance is limited to the priors generated by their modulation network, leaving room for improvement.

Modular approaches provide an interesting avenue for addressing structured problem domains and have achieved some success with solving hierarchical problems in the past [8–11]. An interesting

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

direction in Meta-learning is learning a set of reusable modules from the decomposition of a structured task, and then recombining those modules to solve new tasks [12].

Previous attempts at solving real world multimodal problems have highlighted the inherent structure in many of these problem spaces. For example, various problem domains in the field of robotics are known to be multimodal in nature while maintaining heirarchical structure. An example of this is the task of human action prediction. A popular benchmark dataset for this task is the Berkeley mhad database [13], which includes several data modalities, including accelerometer and video data. Diagnostic tasks in medicine are also often multimodal, combining data from various sensors, tests, text, and imagery to predict the prevalance of a medical condition [14]. In this paper, we propose an extension to modular meta-learning that learns a set of modules and combines them for tasks in a variety of domains. The text is structured as follows: In Section 2, we discuss related work in the areas of modularity, multimodality, and meta-learning. In Section 3, we discuss some critical preliminaries to develop a modular meta learning approach for multiple modalities. Following this, in Section 4 we break down our approach and explicitly outline a pseudocode meta learning algorithm, T4ML. Finally, we discuss experimental aims in detail in Section 5.

2 Related Work

Our work builds primarily from two sources: *multimodal* meta-learning and *modular* meta-learning. Meta-learning empowers machines with the ability of *learning to learn* by designing models that rapidly learn new skills with a few training examples. Notable examples of meta-learning are Model-Agnostic Meta-Learning MAML [3] and related optimization-based methods [4–6, 15]. MAML is a general optimization algorithm that aims to estimate a good initialization of a model's parameters to achieve optimal fast learning on a new task with only a small number of gradient steps. However, having a common initialization for all tasks can restrict the performance on a *multimodal task distribution*.

Multimodal Model-Agnostic Meta-Learning [7] is a more powerful model-agnostic meta-learning framework for the multimodal setting. It augments MAML to identify tasks sampled from a multimodal task distribution and adapts quickly through gradient updates. This framework achieved superior generalization performance in multimodal few-shot regression, image classification, and reinforcement learning tasks.

Recently, investigating the structure of neural networks and designing modular networks has become important for achieving efficient performance. Modularity is an important principle as it provides a natural way of achieving compositionality and generalization, and has been successfully applied to building static neural networks [8, 16–18]. Moreover some studies have found that some types of modular structures emerged in standard neural networks [19, 20]. New strategies have been proposed for combining the modularity of neural networks with meta learning [12, 21, 22], with a general trend of learning modules that can be recombined to solve new tasks, leading to better performance and combinatorial generalization.

We aim to develop a more efficient and adaptable framework that is able to deal with multimodal task distributions while providing modularity and using efficient neural network architectures for new tasks.

3 Preliminaries

As presented in [3], a task is defined by the joint distribution $P_T(x, y)$ over the input, output pairs (x, y). Meta-learning aims to learn functions that approximate mappings for K number of input and output data $(x_k, y_k)_{k=1}^{K_t}$ across a number of different tasks $t \in T$. Data for each task t is split into training and testing datasets, D_t^{train} and D_t^{test}.

MAML: The goal of MAML [3] is to find an initialization of parameters θ for the meta learning algorithm such that convergence to good performance on a new task can be accomplished with relatively few gradient steps trained on D_t^{train} and evaluated for generalization on D_t^{test}. The initialization θ is found by training on groups of tasks and evaluating computed parameters from those tasks to calculate the test losses on the whole test data for the batch of tasks. The gradients of the losses are then used to update θ. We adopt the definition of unimodality and multimodality presented in [7]. If the task distribution contains tasks that belong to a single input domain, it is considered a unimodal distribution. Alternatively, if there are multiple label and input domains, we consider the distribution to be *multimodal*.

Multimodal MAML: Presented in [7], MMAML extends MAML in a framework that allows for learning novel tasks in a multimodal setting. The central idea involves using a modulation network to predict the modality of the task and initialize optimal parameters θ for that modality in the task network. The modulation network works by taking in the input output data $(x_k, y_k)_{k=1}^{K_t}$ and passing them to a task encoder, h. The encoder produces an embedding v, which is then used to compute the task-specific parameters τ that are used to later modulate the meta-learned parameters of the task network. v and τ are formalized as the following:

$$v = h(\{(x_k, y_k)\})_{k=1}^{K_t}; w_h) \tag{1}$$

$$\tau = g_i(v; w_g)_{i=1}^{N} \tag{2}$$

Modulations are practically achieved by applying transformations to each building block in the task network, which is an arbitrarily parameterized function like a Neural Network. Building blocks are denoted by i, and transformations scale and shift the outputs of the neurons in a given block. The function g is made up of feedforward neural networks, each trained to find τ for a single block in the task network. After modulation, the task network undergoes a few steps of gradient descent to achieve optimal performance on the task t.

Modular Meta Learning: In [12], the authors present BounceGrad, an approach that learns a set of modules and combines them to map to new tasks. Starting with a compositional rule and a set of modules, the authors present a hypothesis space defined by the set of functional mappings (C, F, Θ). Modules f in the basis set F are neural networks with varied architectures, parameterized by $\theta \in \Theta$. C corresponds to a compositional scheme for the generation of complex functions from simpler ones, and involves operators that allow for the composition of the neural modules mentioned earlier. $S \in \mathbb{S}$ is a particular structure in the space of all particular structure generated by C through the composition of neural modules f. BounceGrad consists of 2 phases, first learning the optimal structure S and subsequently finding the Θ that minimizes the average generalization performance. During the first phase, Θ is fixed, and the candidate structure S is tested on the training split from a set of Data held out of the main data, known as the meta-test data. The formulation for Phase 1 is as follows

$$S_\Theta^* = \arg\min_{S \in \mathbb{S}} e(D_{meta-test}^{train}, S, \Theta) \tag{3}$$

where $e(D, S, \Theta) = \Sigma_{\{(x,y) \in D\}} L(S_\Theta(x), y)$ defines the loss of a candidate structure with parameters Θ. During the second phase, the structure is now specified, and the goal is to find parameters for the modules that can be used to solve the training tasks. The authors use validation sets for the meta-training tasks to prevent finding parameters that overfit:

$$J(\Theta) = \Sigma_{j=1}^{m} e(D_j^{test}, \arg\min_{S \in \mathbb{S}} e(D_j^{train}, S, \Theta), \Theta) \tag{4}$$

The authors used simulated annealing [23] to search for an optimal structure S, starting with an initial structure and randomly proposing changes that are accepted or rejected. In this work, we propose to extend the BounceGrad approach to the multimodal setting via task aware modulation. We describe our algorithm in the following section.

4 Method

Our goal is to develop a modular approach to multimodal meta-learning through task aware modulation (T4ML). We present an initial T4ML algorithm, which learns a collection of modules that can be combined for a new task samples from a multimodal task distribution. A visual overview of the approach, as well as the complete concrete algorithm in the form of pseudo-code is included in Figure 1 and Algorithm 1, respectively.

T4ML builds on the two phase optimization approach outlined in (3) and (4) by adding a third phase involving task aware modulation. First, after being provided with a basis set of modules F, a compositional scheme C, and an initial set of parameters for the modules Θ, T4ML searches for an optimal structure S^* in the same way as (3). Following BounceGrad, the simulated annealing search is performed starting with an initial candidate structure and randomly modifying it based on the constraints of C to find potentially better candidates. This search is performed over a batch of tasks, as outlined explicitly in lines 3 through 8 of Algorithm 1. Once an optimal structure S^* is found, Phase 2 begins. Θ is now found such that it minimizes the generalization performance of the candidate structure as described in (4). This process is described in lines 9 through 14 of Algorithm 1.

With optimal structure S^* and optimal initilizations Θ found, T4ML introduces task aware modulation as described in MAML. A modulation network is introduced, consisting of a task encoder and a series of deep feed forward neural networks (DNNs). The task encoder generates task embeddings v as described in (1). This is passed to the DNNs, which generate modulation parameters τ as outlined in (2). S^* functions as the task network in the MMAML setting, and task aware initiliazations are found through applying modulations to the module parameters Θ, $\Theta_i \ldots \tau_i$, where $i = 1, ..., N$ for N modules. Different modulations can be used, including softmax based modulation. Different modulation operators will be explored in the experiments outlined in Section 5. During Phase 3, the modulation network and task network are trained end to end on the batch of tasks, as outlined in lines 15 through 23 of Algorithm 1. On a per task basis, j, the gradients for the loss with regards to the data samples K are calculated, and the parameters of Θ are updated. This process is outlined in lines 16 through 19 of Algorithm 1. After training has been done on all tasks in the batch, the parameters are updated for the task network Θ. The parameters for the modulation network are also updated in the same way, denoted by w_h for the task encoder v and w_g for the DNNs that output τ. These updates are described in lines 11 through 13 in Algorithm 1.

We aim to develop and demonstrate the described algorithm, and compare it with state of the art approaches in Modular and Multimodal meta-learning, as described in Section 5.

5 Experimental Approach

To assess the effectiveness of the proposed method in multi-modal settings, we compare it against other meta-learning approaches across multi-modal domains. We adopt and extend the experimental procedures demonstrated in previous meta-learning papers [7, 12] for clarity and reproduceability. T4ML presents an initial exploration into a general approach for modular, multi-modal meta-learning, and as such the methods compared in the experimental approach are specifically those that are known to operate across domains well. Task domains to be explored include multimodal few shot regression and multimodal image classification. These tasks serve as important benchmarks in the broader meta-learning community, and lay the foundation for research into more complicated problem formulations like Reinforcement Learning or Robotics. We seek to compare the following baseline meta-learning methods on the mentioned tasks:

MAML: Model-agnostic meta learners with a fixed task network across different task domains.

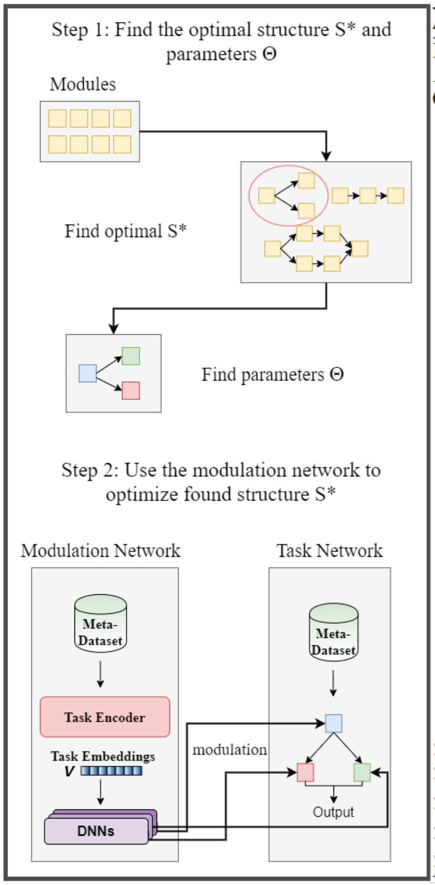
MMAML: The architecture of the task network in the MMAML setup will be identical to MAML.

BounceGrad: The modules in this approach will be shared with T4ML to demonstrate a meaningful baseline for modular approaches that weren't designed with multimodal tasks in mind.

In comparing these distinct methods with T4ML, we hope to highlight the usefulness of modular approaches specialized for multi-modal domains. We outline the Experimental approach for each domain in detail below.

5.1 Regression Domain

We will begin by testing the various baselines on a combination of different function prediction tasks. We follow the general premise introduced in [12] and extend it with the setup of the Sine function prediction task introduced in [3]. We setup 4 different one dimensional functions: sinusoidal functions, linear functions, sums of common non-linear functions, and quadratic functions. Data



Algorithm 1 T4ML

```
Input: $T_0, \Delta_T, T_{end}, S_1 ... S_m,$
$D_1^{Train} ... D_m^{Train}, D_1^{Test} ... D_m^{Test}$
$\Theta$ = network parameter initialization
for $T = T_0; T = T - \Delta_T; T < T_{end}$ do
    $\Delta = 0$
    for $j = 1...m$ do
        $S'_j = ProposeRandom_{S_1...S_m}(S_j, \Theta)$
        $l' = e(D_j^{Train}, S'_j, \Theta)$
        $l = e(D_j^{Train}, S_j, \Theta)$
        if $Accept(l', l, T)$ then
            $S_j = S'_j$
        end if
    end for
    for $j = 1...m$ do
        $(x, y) = RandomElement(D_j^{test})$
        $\Delta = \Delta + \nabla_\Theta L(S_{j_\Theta}(x), y)$
    end for
    $\Theta = \Theta - n\Delta$
end for
With optimal $S$ and associated parameters $\Theta$
Initialize $w$ randomly for modulation network
while not DONE do
    for $j = 1...m$ do
        $v = h(\{x, y\}_K; w_h)$
        $\tau = \{g_i(v; w_g | i = 1, ..., N\}$
        $L = L_{T_j}\left(f(x; \Theta, \tau); D_{T_j}^{train}\right)$
        $\Theta'_{T_j} = \Theta - \alpha \nabla_\Theta L_{T_j}(f(x; \Theta, \tau); D_{T_j}^{train})$
    end for
    Update $\Theta$ with $\beta \nabla_\Theta \Sigma_{T_j} L_{T_j}(f(x; \Theta', \tau); D_{T_j}^{test})$
    Update $w_g$ with $\beta \nabla_{w_g} \Sigma_{T_j} L_{T_j}(f(x; \Theta', \tau); D_{T_j}^{test})$
    Update $w_h$ with $\beta \nabla_{w_h} \Sigma_{T_j} L_{T_j}(f(x; \Theta', \tau); D_{T_j}^{test})$
end while
```

Figure 1: **Left:** T4ML Overview. Step 1 involves finding optimal network structure S^* and initialization Θ. Step 2 modulates S^*_Θ using the generated parameters τ. **Right:** T4ML algorithm.

is samples with gaussian noise added to the output values. Pairs of input and output samples will be sampled from the function being tested and passed to the different meta-learning baselines for learning. Hyperparameters of established methods like MMAML and BounceGrad will be chosen based on their optimal configuration as provided in their respective papers for regression experiments. Hyperparameters for T4ML will be explored using similar initial values as seeds for a more general grid search. The model is tasked with predicting output values for associated inputs. We will repeat the experiments across 10 different runs and ascertain statistical significance through variance estimates of the performance for each of the methods, with the general aim of receiving experimental results that are conclusive.

Method Configurations MAML and MMAML will both make use of deep feedforward neural networks as task networks. MMAML will use an LSTM as the modulation network because of its success with sequential inputs and its use in the original MMAML configuration. As compared to the previous baselines, BounceGrad and T4ML make use of a compositional structure to construct the equivalent of the task network in MMAML. The general compositional scheme for sinusoidal functions, linear functions, and quadratic functions will be $h(x) = f_i(f_j(x))$ and $h(x) = f_i(x) + f_j(x)$ for sums of non-linear functions, following the general setup of experiments in [12]. The compositional space F consists of 10 feedforward neural network modules, half of which have 1 hidden layer and half of which have 2. T4ML makes use of the same LSTM based modulation network as MMAML. Modulation approaches will include FiLM [24] and softmax [25].

5.2 Image Classification

For Multimodal Image Classification, we generally follow the experimental procedure set forth in [7]. The task can be summarized as classifying images into a set of classes with a few number of samples available. We will combine several popular image datasets into a multimodal few-shot image dataset. The datasets to be used for this task are Mini-ImageNet [26], FC100 [27], CUB [28], AIRCRAFT [29], and OMNIGLOT [30]. classification Models are trained on different combinations of modalities, mainly 3 and 5 mode combinations. This procedure follows the some of the experiments presented in the original MMAML paper, and allows for a direct comparison between T4ML and MMAML. As with the regression experiments, we will repeat the experiments across 10 different runs in order to observe statistically significant results. This pattern will be repeated for all models being compared in the experiments.

Method Configurations In these experiments, MAML and MMAML will now make use of small 5 layer convolutional neural networks, along with the same LSTM modulation network. The compositional scheme used by BounceGrad and T4ML will be $h(x) = f_i(f_j(x))$ owing to the heirarchical nature of the image domain, and the compositional space will consist of CNN modules, with 3 and 5 layer combinations. T4ML will continue to use an LSTM for task aware modulation, and will use FiLM and softmax.

References

[1] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning; On learning how to learn: The meta-meta-...hook.* PhD thesis, Institut f. Informatik, Tech. Univ. Munich, 1987.

[2] S. Thrun and L. Pratt. *Learning to learn.* Springer, 1998.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning (ICML)*, 2017.

[4] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[5] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *InInternational Conference on Learning Representations Workshop*, 2018.

[6] Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

[7] Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Advances in Neural Information Processing Systems*, pages 1–12, 2019.

[8] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48, 2016.

[9] Gasser Auda and Mohamed Kamel. Modular neural networks: a survey. *International Journal of Neural Systems*, 9(02):129–151, 1999.

[10] Bart LM Happel and Jacob MJ Murre. Design and evolution of modular neural network architectures. *Neural networks*, 7(6-7):985–1004, 1994.

[11] Harshvardhan Sikka. Creating, managing, and understanding large, sparse, multitask neural networks. 2020.

[12] Ferran Alet, Tomás Lozano-Pérez, and Leslie P Kaelbling. Modular meta-learning. *Proceedings of The 2nd Conference on Robot Learning*, 2018.

[13] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 53–60. IEEE, 2013.

[14] Yong Xia, Zexuan Ji, Andrey Krylov, Hang Chang, and Weidong Cai. Machine learning in multimodal medical imaging, 2017.

[15] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 7332–7342, 2018.

[16] Michael B Chang, Abhishek Gupta, Sergey Levine, and Thomas L Griffiths. Automatically composing representation transformations as a means for generalization. *In International Conference on Learning Representations*, 2019.

[17] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: what is required and can it be learned? *In International Conference on Learning Representations*, 2019.

[18] Louis Kirsch, Julius Kunze, and David Barber. Modular networks: Learning to decompose neural computation. In *Advances in Neural Information Processing Systems*, pages 2408–2418, 2018.

[19] Chihiro Watanabe, Kaoru Hiramatsu, and Kunio Kashino. Modular representation of layered neural networks. *Neural Networks*, 97:62–73, 2018.

[20] Daniel Filan, Shlomi Hod, Cody Wild, Andrew Critch, and Stuart Russell. Neural networks are surprisingly modular. *arXiv preprint arXiv:2003.04881*, 2020.

[21] Rohan Chitnis, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Learning quickly to plan quickly using modular meta-learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7865–7871. IEEE, 2019.

[22] Yutian Chen, Abram L Friesen, Feryal Behbahani, Arnaud Doucet, David Budden, Matthew W Hoffman, and Nando de Freitas. Modular meta-learning with shrinkage. *NeurIPS Workshop on Meta-Learning*, 2019.

[23] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[24] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *In AAAI*, 2018.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[26] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[27] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, pages 721–731, 2018.

[28] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[29] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[30] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.